Light Indexed Deferred Rendering

Yanni Coroneos

First by Damian Trebilco

Rendering in Linear Time

- Forward Rendering:
  - Render every object for every light
  - O(objects * lights) <-SLOW
- Light Indexed Deferred
  - Render light map then use it later to shade every object
  - O(lights) + O(objects) <-Much Better

Overview

1. Assign each light an index between 0 and 255
2. Render depth pre-pass
3. Figure out which fragments are hit by each light using implicit light volumes and generate a “light map”
4. Pack light properties into 1D textures
5. Draw geometry in a forward pass and use the light map to find which light hit the fragment. Then use the indices to look up properties from the textures.

Overview

1. Assign each light an index between 0 and 255
2. Render depth pre-pass
3. Figure out which fragments are hit by each light using implicit light volumes and generate a “light map”
4. Pack light properties into 1D textures
5. Draw geometry in a forward pass and use the light map to find which light hit the fragment. Then use the indices to look up properties from the textures.

Read the paper

https://github.com/ycoroneos/LightDraw

1024x768
lidr no shadow no vsync
lidr shadow no vsync
forward no shadow
forward shadow
FPS
lights
0
100
200
300
400
500
0
50
100
150
200
250

1920x1080
lidr no shadow no vsync
lidr shadow no vsync
forward no shadow
forward shadow
FPS
lights
0
100
200
300
400
500
0
50
100
150
200
250

3840x2160
lidr no shadow no vsync
lidr shadow no vsync
forward no shadow
forward shadow
FPS
lights
0
100
200
300
400
500
0
50
100
150
200
250

Video

- Things I didn’t talk about
  - Game engine
  - Bit packing procedure
  - Memory advantages of LIDR
  - How I benchmarked / interpreted results
  - Comparison to other deferred rendering techniques